BirdCLEF+ 2025

Haochen Li, Haoran Cui, Will Weng, Gary Wang

[**Abstract**](#_ln6a7lgir6f) **3**

[**Introduction**](#_tu0l1qvht7r4) **3**

[**Data**](#_yewqg6bniqc2) **4**

[**EDA**](#_kas9iuvsrjds) **5**

[Data structure](#_45i7jhunenzv) 5

[Audio Characteristics](#_eftg6nwak1ur) 8

[**Methodology**](#_5j3ku01l35vg) **10**

[Feature engineering](#_cmd1eonlt6eg) 10

[**Model**](#_lcsluyehxc3j) **13**

[Convolutional Neural Network](#_588xdi5kiw95) 13

[EfficientNet Family](#_ksrzggrgfxfe) 14

[EfficientNetB0](#_s627b944t50o) 15

[EfficientNetV2](#_mr9e8vyhfsac) 15

[**Result**](#_8uz0a9r8he6) **17**

[**Discussion**](#_g1ou8ata459j) **18**

[**Reference**](#_4m52857ocfsj) **19**

# 

# **Abstract**

*In the BirdCLEF 2025 challenge, we developed a scalable pipeline to detect bird species from audio soundscapes using deep learning. Raw recordings were denoised, segmented, and converted into mel spectrograms. These spectrograms were fed into ensemble models based on EfficientNetB0 and EfficientNetV2, with test-time augmentation and temporal smoothing to improve robustness. We performed inference on over 9,000 soundscape recordings, generating more than 100,000 prediction rows. Despite challenges like severe class imbalance, overlapping vocalizations, environmental noise, and domain shift between clean training clips and noisy multi-label test audio, the system produced stable outputs in a real-world setting. Cross-validation results showed EfficientNetV2 achieved higher stability and faster convergence.*

# **Introduction**

Mobile and habitat-diverse species serve as valuable indicators of biodiversity change, as shifts in their assemblages and population dynamics can signal the success or failure of ecological restoration efforts. In this case, tracking shifts in their acoustic signatures is a wise idea to examine whether restoration is on the right track or drifting off-course, especially for those endangered species. However, traditional observing methods can hardly keep pace with the scale of such a task in the Magdalena Valley, where the forest sprawls across thousands of acres and the sound changes hour by hour. Although researchers have installed Passive Acoustic Monitoring (PAM) recorders, the raw audio still needs to be processed and translated by algorithms in order to gain any insightful information.

This is exactly what the BirdCLEF 2025 project aims to accomplish. Our goal is to build an efficient pipeline that is able to extract clips from 24-kHz streams and turn them into spectrogram images. Then, stripping out irrelevant noises and using machine learning techniques like CNN and EfficientNet to identify visual patterns in the images and learn which labels go with which bird. Last but not least, performing 10-fold cross-validation to examine the model’s performance and accuracy.

# **Data**

The data provided contains both the desired single-species calls for supervised learning and the complex soundtracks that require further analysis.

The foundation of the project is a folder named “train\_audio,” which contains a library of single-species recordings that have all been resampled at 32 kHz and converted to Ogg format for consistency. Each clip in this folder is paired with rich row-level metadata in “train.csv” that serves as the main manifest. It links every recording with its primary label, any co‑occurring species, geographic coordinates, recorder information, and quality ratings. In addition, this dataset is complemented by “taxonomy.csv,” which maps every kind of animal to its species, such as aves and amphibians, so that class‑balanced sampling and hierarchical loss functions can be applied if necessary.

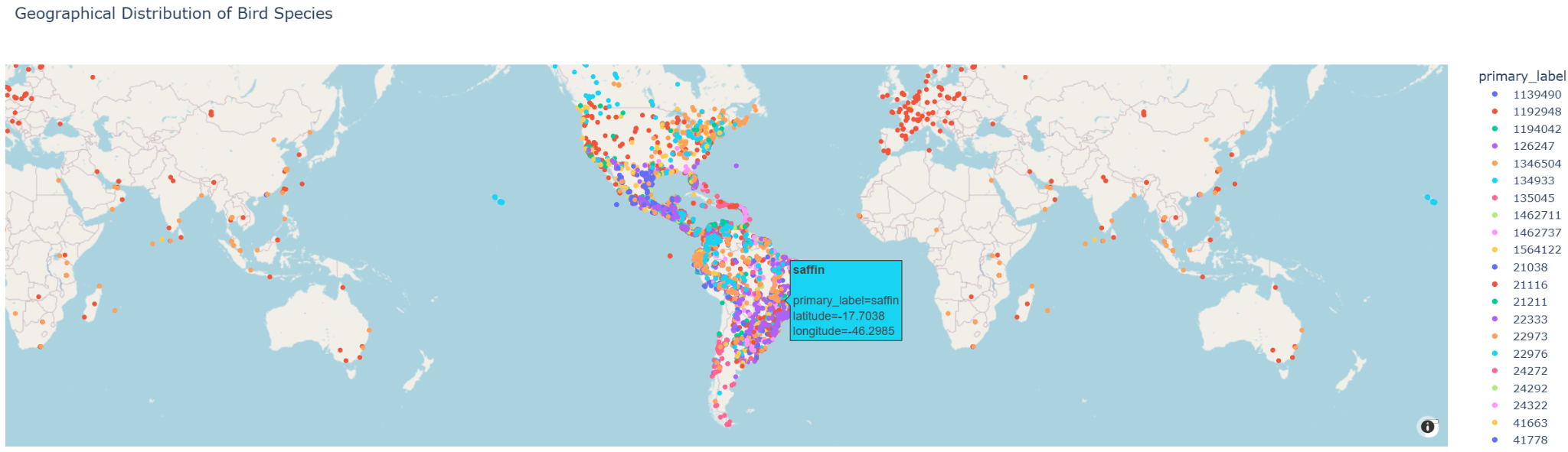
On the other hand, “train\_soundscapes” contains longer, unlabeled environmental recordings from the same forest. Although recorded at the same location, precise recording sites of unlabeled soundscapes do not overlap with recording sites of the hidden test data, “test\_soundscapes.” At the time of submission, Kaggle will use around 700 one-minute random soundscapes for scoring.

In total, these files yield 28,564 individual observations. This ensures us a dataset that is both large enough to train models like CNN and perform cross-validation to keep our pipeline reproducible.

# **EDA**

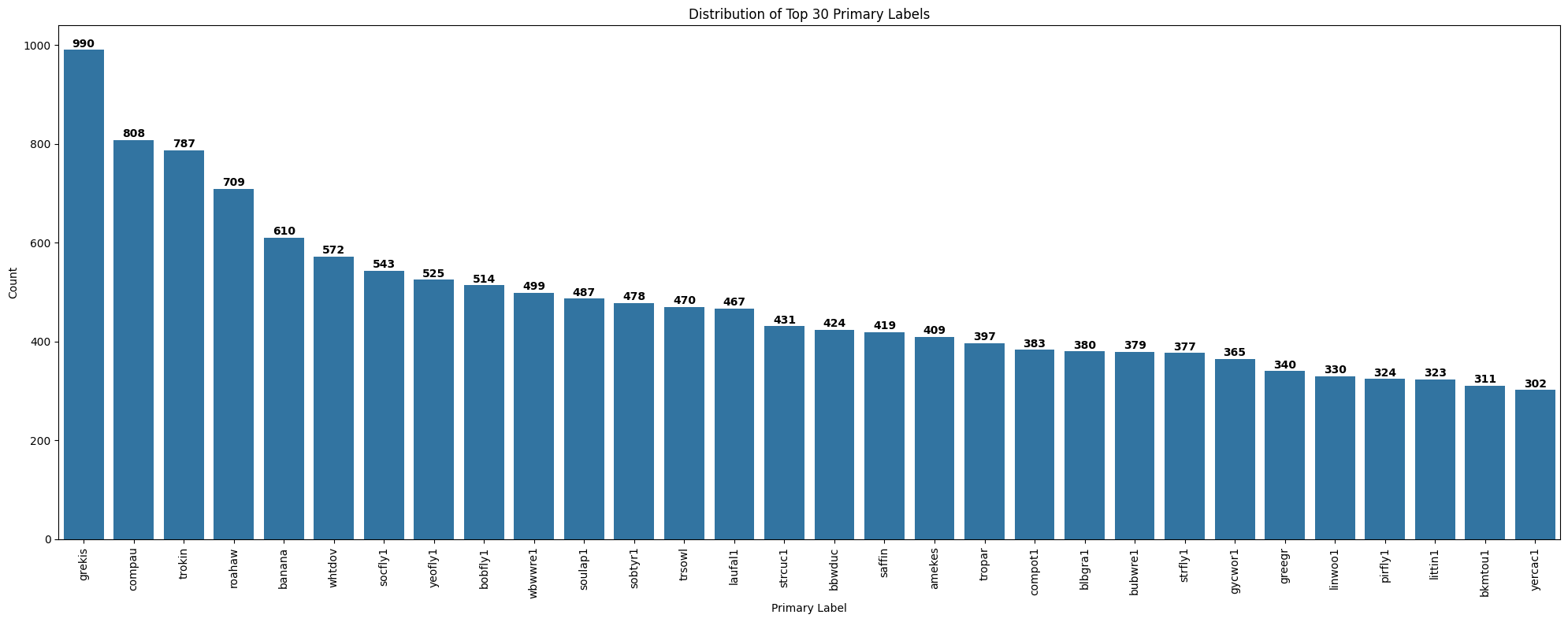
In our exploratory analysis, we aimed to uncover patterns, potential biases, and key data characteristics essential for developing an effective bird species classification model based on audio recordings.

We began by examining the geographical distribution of the dataset. Most recordings originated from South America, revealing a regional concentration in data collection. This geographic skew highlights the importance of accounting for location-based biases. We recognize the need to incorporate more recordings from underrepresented regions to enhance the generalizability and robustness of our classification model.

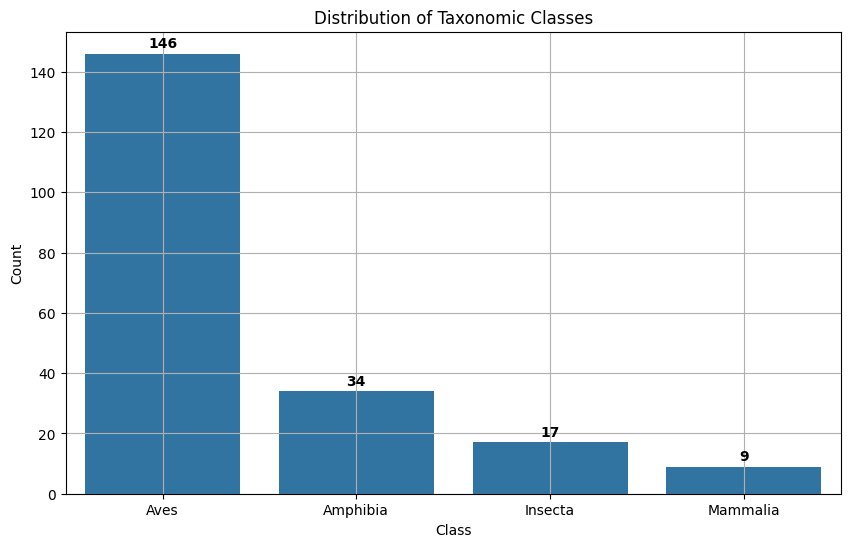


Next, we assessed the frequency distribution of species labels, which revealed a pronounced class imbalance. The most frequent species had up to 990 recordings, while many others had far fewer. This imbalance can bias model predictions toward dominant classes. We also observed this pattern in the distribution of vocalization types. Certain species consistently dominated specific behaviors, such as songs and calls, with some contributing over 20% of the total samples in those categories. To mitigate this issue, we considered balancing strategies and class weighting in our model design.

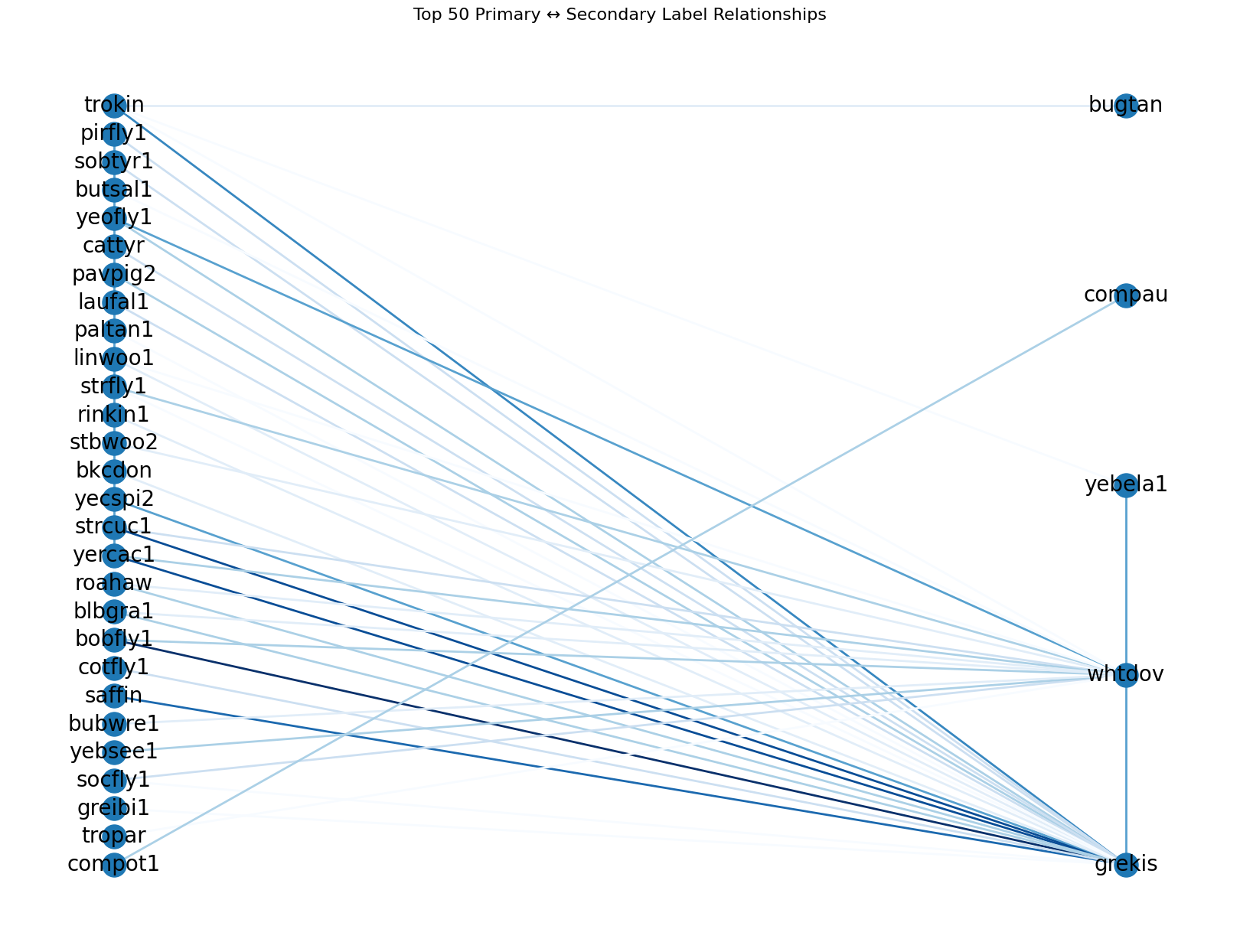
## **Data structure**



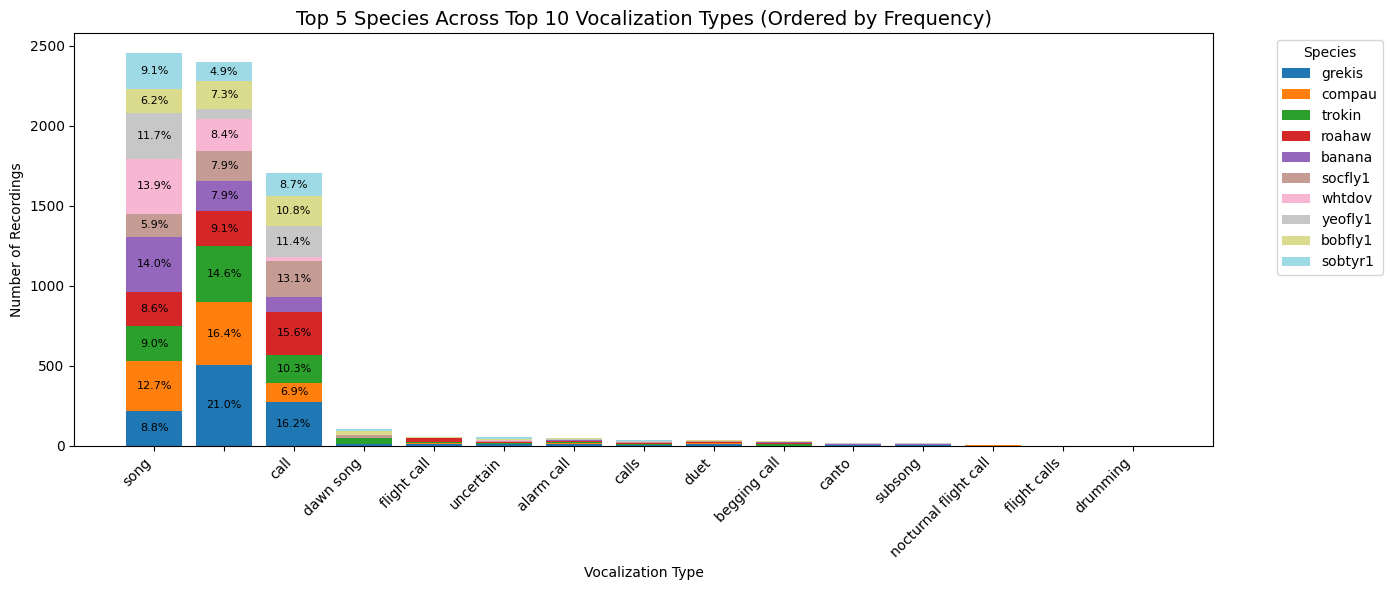
We further explored the taxonomic composition of the dataset. Our findings showed that 'Aves' was the most represented class with 146 unique labels, compared to only 34 for 'Amphibia', 17 for 'Insecta', and 9 for 'Mammalia'. This imbalance across taxonomic classes led us to implement stratified sampling during cross-validation to maintain representative distributions across training and validation sets.



Analyzing primary and secondary label relationships, we observed frequent co-occurrences that suggest underlying acoustic similarities among species. These complex interdependencies indicate that a multi-label or hierarchical modeling approach may better capture the nuances of species classification based on overlapping vocal patterns.

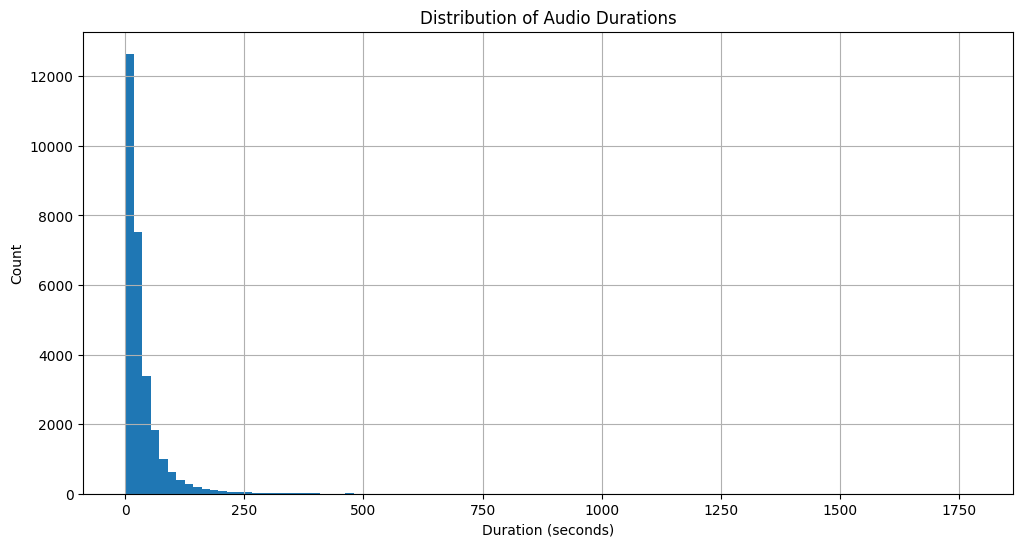


We also investigated vocalization types across leading species. 'Song' and 'call' dominated the dataset, accounting for the majority of samples, which can benefit model training by offering consistent patterns. However, rare vocalizations were sparsely distributed and presented challenges due to their limited representation. We plan to address this by incorporating data augmentation techniques and specialized feature engineering.

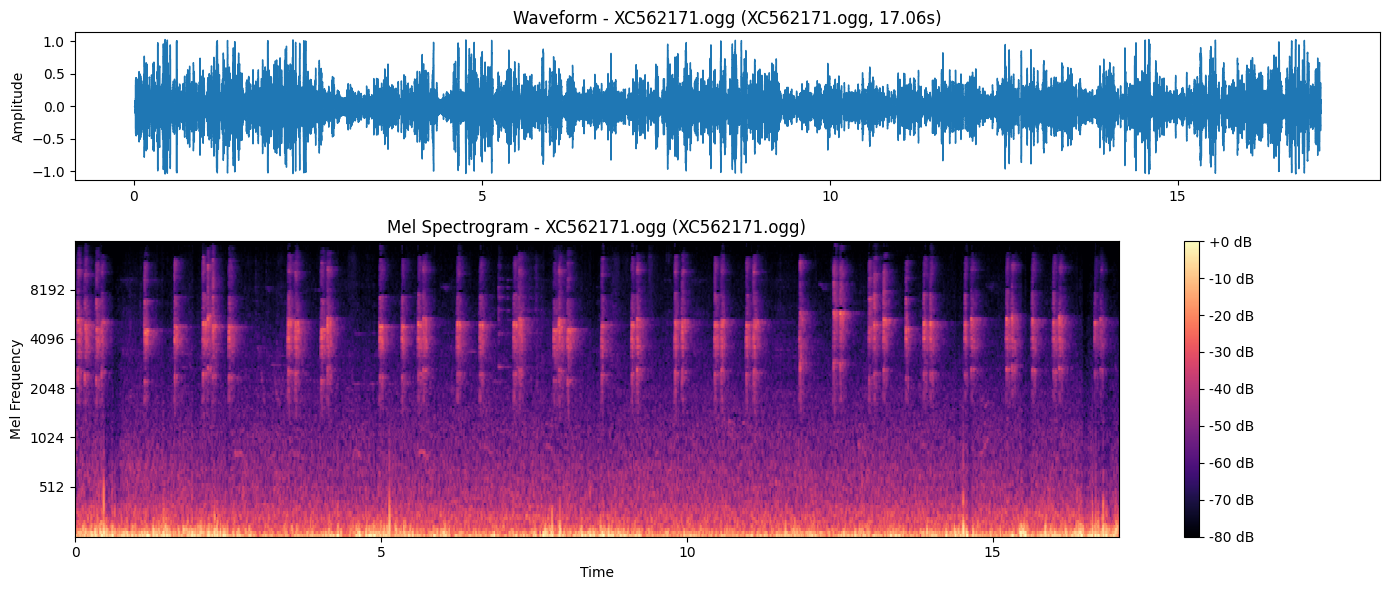


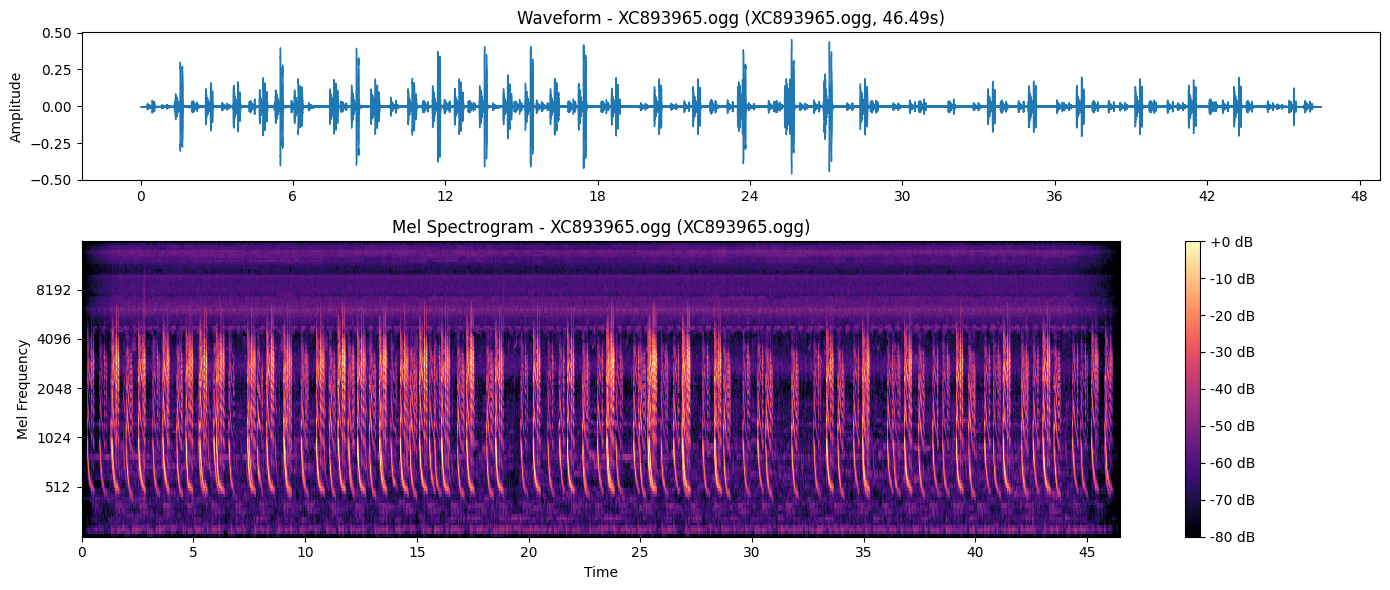
## Audio Characteristics

Our analysis of audio durations revealed substantial variability, with durations ranging from 0.54 seconds to 1774.39 seconds. The mean duration was 35.35 seconds, while the median was 20.98 seconds, confirming a right-skewed distribution. Approximately 15% of recordings exceeded 60 seconds, while a small fraction (0.08%) were shorter than 1 second. To handle this variability, we applied segmentation and adaptive padding techniques to standardize input lengths for modeling.



To evaluate signal characteristics, we visualized waveform plots alongside Mel spectrograms. The waveform offered a direct view of amplitude variation over time, helping us identify silent periods, signal density, and envelope shapes. When translated into the Mel spectrogram, these temporal patterns became frequency-rich visualizations that highlighted harmonic structures, repetition, and silence with greater clarity. For example, a waveform with periodic high-amplitude bursts corresponded to distinct vertical striations in the Mel spectrogram, reflecting species-specific vocal pulses. This direct correspondence between time-domain and frequency-domain characteristics validated our choice to rely on Mel-frequency features for model input, as they encapsulate both spectral texture and temporal rhythm, crucial for distinguishing bird calls.

****

****

# Methodology

## Feature engineering

Feature engineering (FE) refers to the process of transforming raw data into meaningful representations that enhance a model's ability to learn predictive patterns. In the context of this project, where the objective is to classify bird species from audio recordings, feature engineering is crucial for extracting informative characteristics from variable-length, noisy, and weakly labeled sound data. The raw audio files themselves are not directly suitable for training deep learning models due to their high dimensionality, temporal variability, and inconsistent quality. Therefore, careful preprocessing and feature transformation are necessary to convert these inputs into structured formats that are both computationally efficient and biologically meaningful. As part of FE, we also extracted statistical features such as mean frequency, spectral centroid, and bandwidth to explore their impact on model performance. However, the primary features used were image-based spectrograms, which aligned best with CNN architectures

All audio files were first loaded and resampled to a uniform sample rate of 32,000 Hz to ensure consistency across the dataset. Since recordings varied in duration, each audio clip was standardized to a fixed 15-second length. Short clips were zero-padded, while longer ones were truncated. This fixed duration corresponded to a total of 480,000 samples per audio file (15 \* 32000), which was necessary for producing spectrograms of uniform width (384 pixels). This standardization facilitated downstream FE by ensuring that all resulting spectrograms had consistent time-frequency dimensions, which simplified batching and model input shaping.

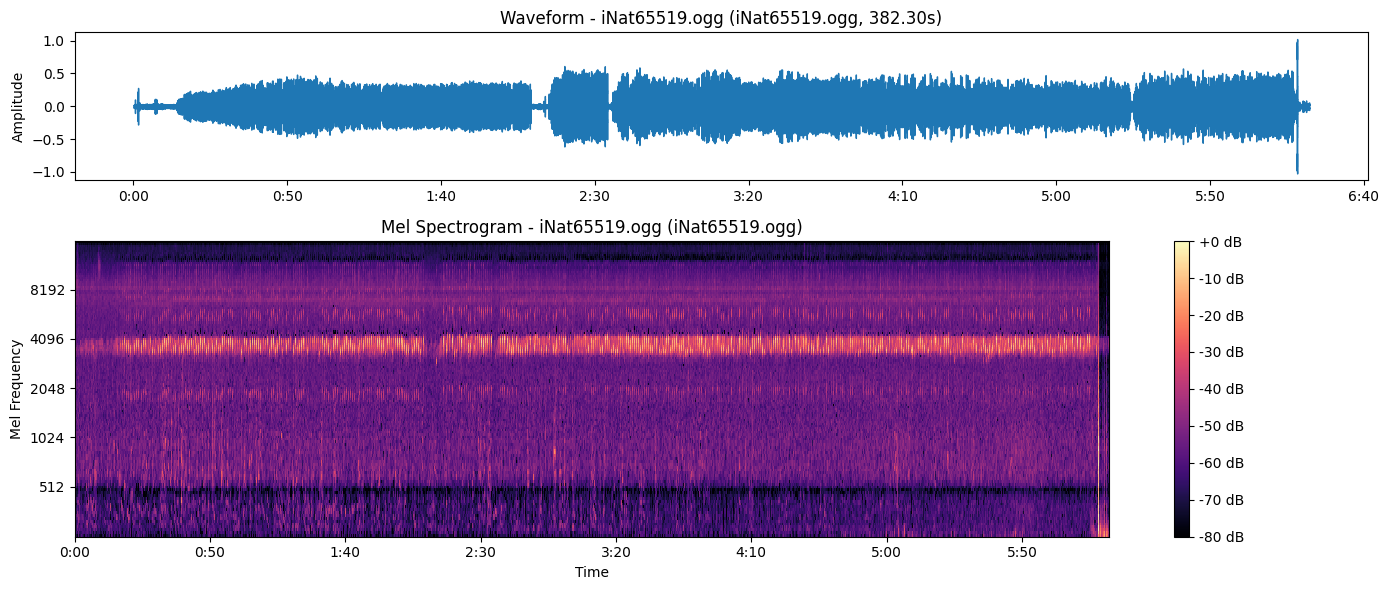
Following temporal alignment, a Biodenoise module was applied to each waveform after resampling and padding. The raw audio was resampled to a uniform 32,000 Hz and padded or truncated to 15 seconds (480,000 samples). The denoising step preceded any spectral transformation and was conducted in no-gradient inference mode for computational efficiency. This model-based denoising method was designed to suppress ambient noise sources such as wind, insects, or human-made sounds, which commonly obscure subtle vocalizations in bioacoustic data. By removing these non-biological interferences at the waveform level, the audio signal was acoustically purified before feature extraction.

After denoising, the cleaned waveform was used to compute a mel spectrogram, which transformed the temporal signal into a time-frequency domain. A 2,028-point short-time Fourier transform (STFT) with a 1,024-sample Hann window and a hop length tailored to yield a 384-pixel time axis was applied. The mel spectrogram comprised 128 mel bands covering a frequency range from 20 Hz to 16,000 Hz, aligned with typical bird vocalization frequencies. This choice of parameters for STFT and mel bands was part of the feature engineering design, optimized through preliminary experimentation to balance resolution with computational cost.

To enhance the contrast between signal and silence, the power spectrogram was converted to the decibel (dB) scale. Finally, a min-max normalization was performed by subtracting the minimum and dividing by the range, scaling the spectrogram values to the [0, 1] interval. This normalization mitigated differences in absolute energy across recordings and ensured that inputs were on a consistent scale for deep learning models. It also improved model convergence and allowed the network to prioritize relative intensity patterns, which are more robust indicators of species-specific vocalizations than raw amplitude levels.

Following this, the power spectrogram was converted into the decibel (dB) scale to enhance contrast between silent and active regions. The dB-scaled spectrogram was then min-max normalized: its minimum value was subtracted, and the result was divided by the maximum, scaling the values to the [0, 1] range. This normalization ensured that pixel intensity values across spectrograms were on a consistent scale, which is essential for stable training of convolutional neural networks. It also mitigated the influence of absolute amplitude, allowing the model to focus on relative energy patterns associated with vocalizations.

In the subsequent step, we computed the mel spectrogram for each audio file, we split each dataset into 5-second chunks, and stored them in .png format. Each file included grey scale and converted to RGB bands and matched the format requirements for training convolutional neural networks. This chunking process supported data augmentation and enabled the model to process long recordings more efficiently by treating each short segment as a standalone training instance.



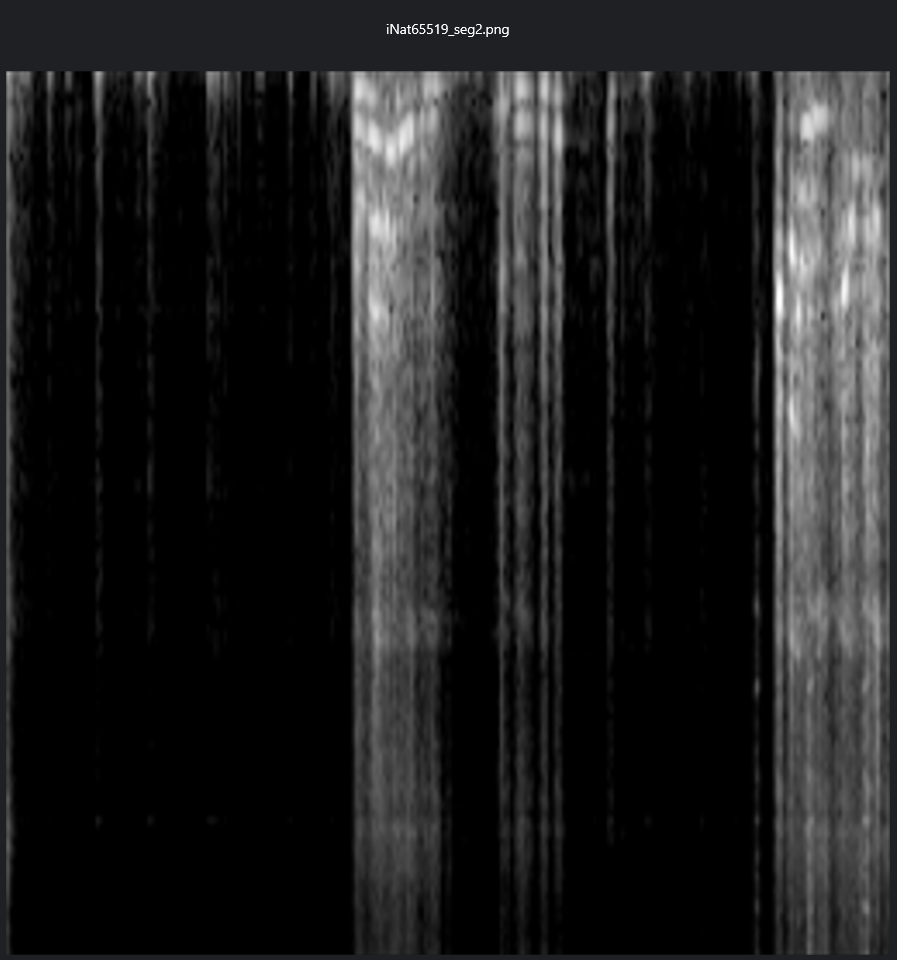
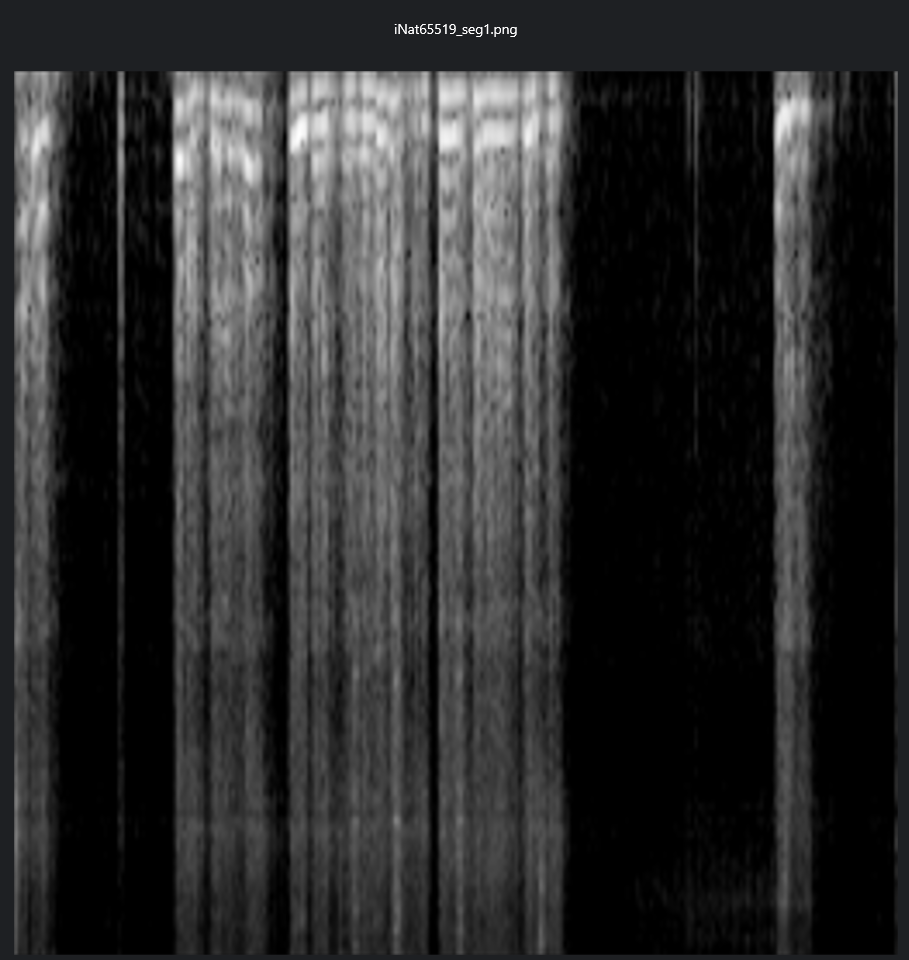
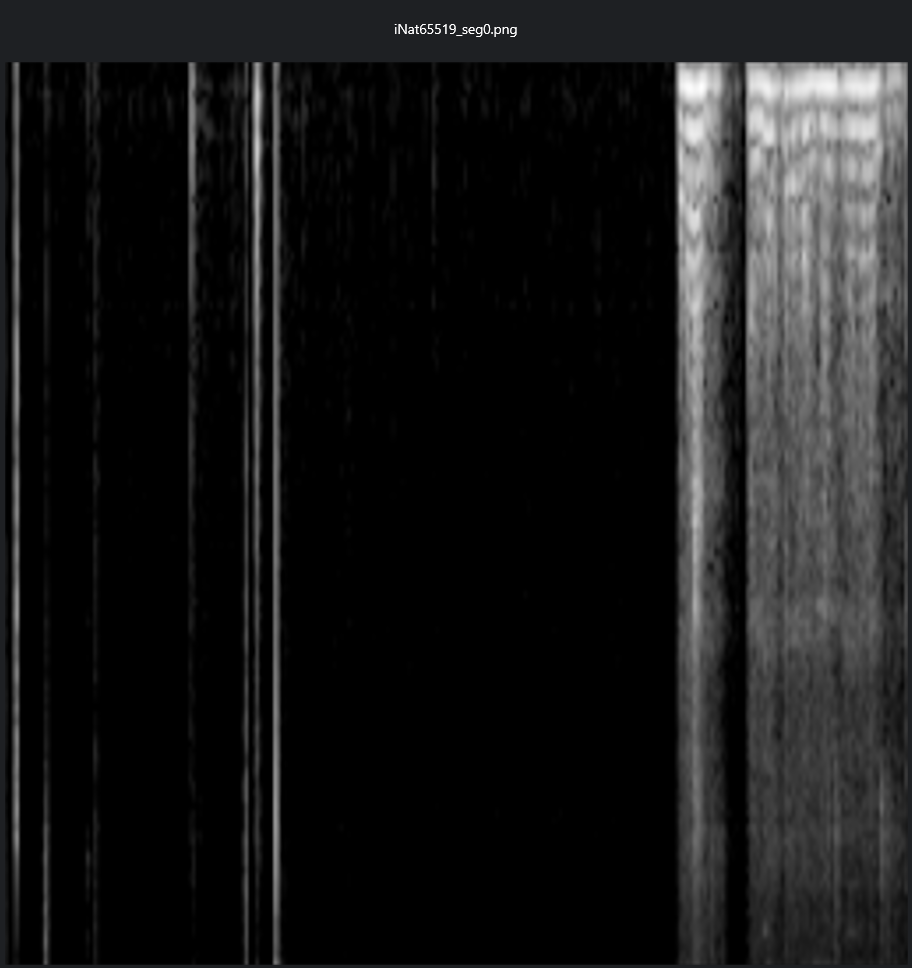


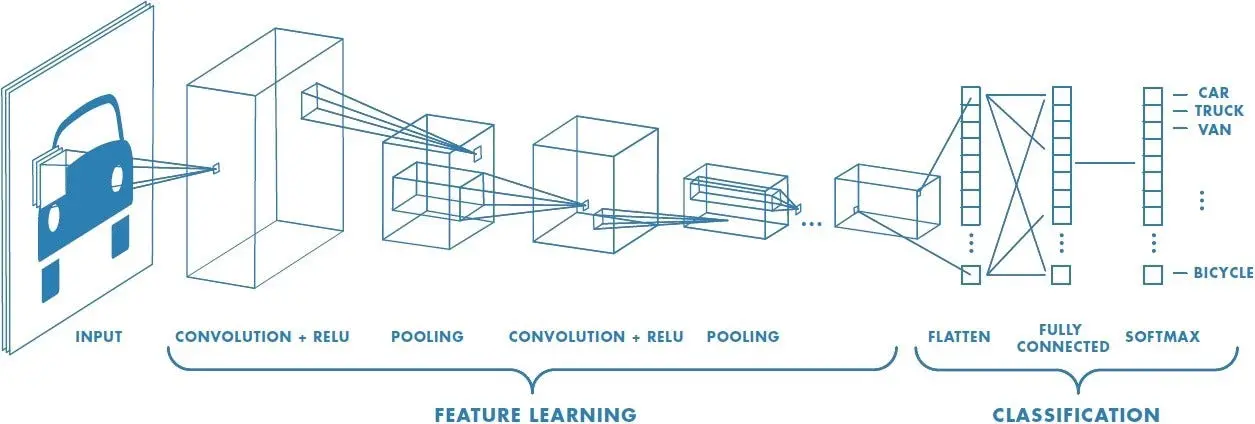
Fig. From left to right, is iNat65519\_seg0.png, iNat65519\_seg1.png, iNat65519\_seg3.png. 5 second each image.

To support efficient supervised learning, a custom PyTorch Dataset class was developed to load these spectrogram images and assign corresponding species labels. The class parsed filenames to extract species identifiers, which were then mapped to integer class indices using a label dictionary constructed from the dataset. A costumed dataset ensured the reproducibility and scalability while enabling integration with standard data loaders for mini-batch training, optional image augmentation, and real-time preprocessing. Together, these steps established a structured, high-throughput pipeline suitable for deep learning in the context of large-scale bioacoustic classification. The integration of image transformations during data loading also falls under the umbrella of online feature engineering, as it dynamically enhances the training set without modifying the original files.

# **Model**

## Convolutional Neural Network

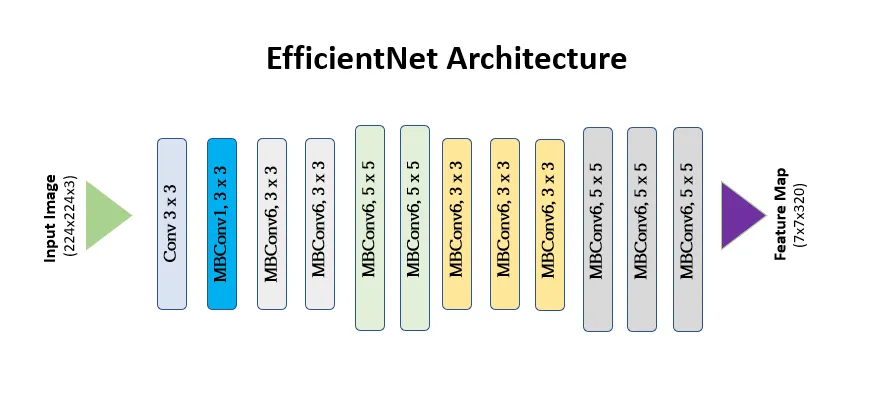
Convolutional Neural Networks (CNNs) are purpose‑built for recognising patterns that repeat locally within a grid. It can identify edges in photographs, strokes in handwriting, but also the ridges and harmonics that appear in an acoustic spectrogram. A spectrogram is simply an image where time runs along the horizontal axis, frequency along the vertical, and pixel intensity encodes the power of the signal at each time or frequency coordinate. Because bioacoustic events such as bird trills, frog croaks, and insect stridulations form compact, characteristic shapes within that grid, a CNN can learn filters that light up whenever those shapes appear, regardless of where they fall in the recording. Convolutional layers scan the spectrogram with small learnable filters that detect local features, while pooling layers reduce the spatial resolution, which helps the network generalize across time and frequency shifts. After that, fully connected layers integrate the condensed feature maps into a final prediction, mapping the learned acoustic patterns to the correct species label. As a result, the model can correctly recognize a species even if the key sound appears early or late in the audio clip.



In our project, each 15-second audio sample is first resampled to 32000 Hz and transformed into a log-mel spectrogram with 128 mel bands and a hop length of around 12 milliseconds. The resulting spectrogram is normalized and saved as a PNG image with a size of 128 by 384. This approach decouples audio processing from model training, which allows us to quickly load precomputed spectrograms during training. We built a custom dataset class that reads these images, extracts labels from filenames, maps them to integer IDs, and applies basic image transformations such as resizing to 224 by 224 pixels and converting to tensors.

We then used a pretrained ResNet18 model as the backbone for classification. All convolutional layers are kept as they are, while the final classification layer is replaced with a new fully connected layer that matches the number of target species. This transfer learning setup allows the network to leverage general image features learned from ImageNet and adapt them to our spectrogram data. Training also used the Adam optimizer with a learning rate of 0.0001 and a batch size of 64. Even after only three training epochs, the model achieved around 70 percent accuracy on a validation split, which shows that CNNs are well-suited for this task.

## EfficientNet Family

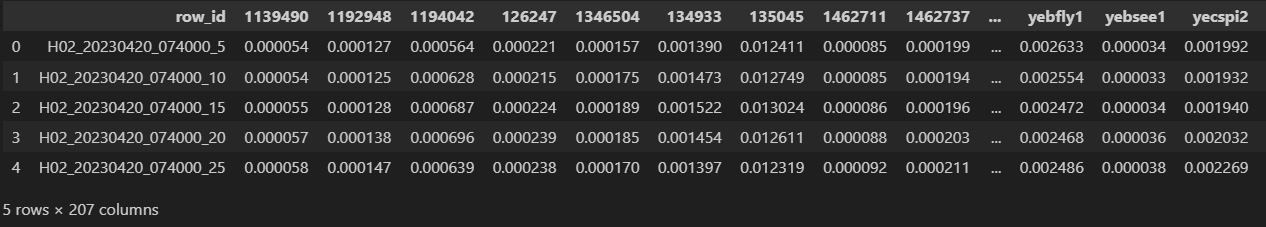


After we performed CNN for baseline, we employed two convolutional neural network architectures from the **EfficientNet family**: **EfficientNetB0** and **EfficientNetV2**. These models are known for their optimal balance of performance and computational efficiency, making them ideal choices for audio-based image classification tasks like ours.

## EfficientNetB0

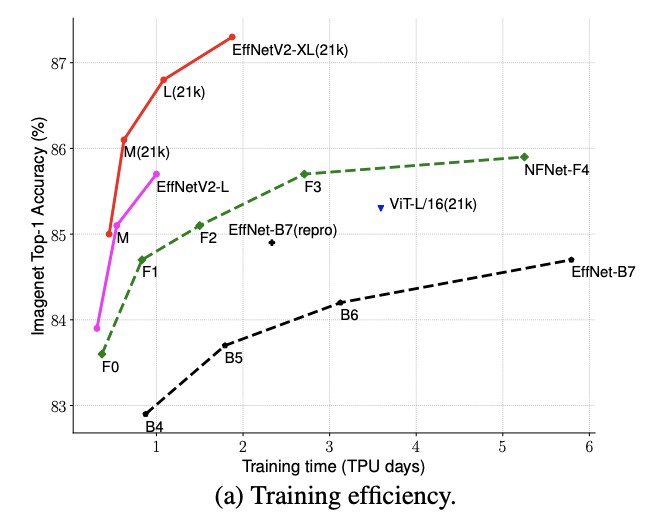
EfficientNetB0 is the baseline model in the original EfficientNet architecture proposed by Tan and Le (2019). It introduces a novel **compound scaling method** that uniformly scales a network’s depth, width, and resolution using a set of fixed coefficients. This approach allows the model to maintain a high level of accuracy while significantly reducing the number of parameters and FLOPs (floating point operations) compared to traditional CNNs like ResNet or Inception.

EfficientNetB0 uses **mobile inverted bottleneck (MBConv)** blocks and **squeeze-and-excitation (SE)** modules to model inter-channel dependencies and enable efficient feature learning. Its lightweight structure is particularly advantageous for training on modest hardware or deploying in resource-constrained environments. In our context, EfficientNetB0 was used as a strong baseline to process mel spectrograms, offering fast inference and solid accuracy on lower-resolution inputs.



## EfficientNetV2

EfficientNetV2 is a modern convolutional neural network architecture that specializes in improving both accuracy and training speed over earlier models like the original EfficientNet. Like other CNNs, EfficientNetV2 operates on grid‑structured inputs such as spectrograms, which use stacks of convolutional layers to detect spatial patterns. However, what sets EfficientNetV2 apart is its compound scaling method and updated architecture that uses a mix of mobile inverted bottleneck convolution (MBConv) blocks and fused MBConv blocks to accelerate training and improve performance, especially on smaller datasets and lower-resolution inputs. Compared to the original EfficientNet, EfficientNetV2 is more efficient in both compute and memory use. It drops some of the computationally expensive squeeze-and-excitation layers in early stages and replaces them with faster fused convolutions, which combine a standard convolution and batch normalization into a single step. As a result, EfficientNetV2 models train significantly faster while maintaining or even improving classification accuracy. This makes them well-suited for scenarios like ours, where we deal with medium to low resolution spectrograms and want quick iteration cycles.



In our project, EfficientNetV2 was integrated as an upgrade path from the baseline ResNet18 model. Within our configuration class, we defined the preferred model preset as 'efficientnetv2\_b2\_imagenet.' While the training loop currently demonstrated the pipeline using ResNet18 for simplicity, the rest of our code was modular and can easily switch to EfficientNetV2 by updating the model initialization block. This design choice allowed us to preserve all data preprocessing and loading routines, while swapping in a more powerful backbone when training at scale.

# Result

We ran inference on over 9,000 audio files from the train\_soundscapes folder to simulate real-world performance. Each file was segmented into overlapping 5-second chunks. Segments shorter than 3.75 seconds were discarded. On average, each file contributed more than 10 valid segments. This resulted in a total of 116,712 raw prediction entries, out of which 106,986 valid rows were retained after filtering.

Each 5-second segment was transformed into a mel spectrogram. Denoising and bandpass filtering were applied first to reduce ambient forest noise and remove irrelevant frequencies. Spectrograms were normalized and fed into an ensemble of EfficientNet-based models (B0 and V2 variants). Each model returned sigmoid-based probability scores for 182 bird species.

The final output was saved in sample\_submission\_local\_train.csv, matching the expected submission format. Each row included a unique row\_id and corresponding 182 probability columns. The majority of scores were close to 0.01, suggesting the model was cautious in assigning high probabilities unless a confident match was found. This is expected in a multi-label soundscape environment where true positives may be sparse.

To improve robustness, we applied test-time augmentation (TTA). Each spectrogram was flipped along the time and frequency axes and intensity-adjusted. Predictions from all augmentations were averaged to stabilize outputs. Additionally, predictions were smoothed temporally. For each row, we averaged probabilities with nearby segments using a weighted moving average. This reduced jitter in detection, especially for species with intermittent calls.

We also evaluated model training performance using 5-fold cross-validation. EfficientNetB0 reached an average validation accuracy of 72.1%, showing good generalization and low variance across folds. EfficientNetV2 converged faster and performed slightly better under noisy conditions, with more stable log-loss curves. Training curves showed no significant overfitting, thanks to regularization from augmentation, denoising, and normalization.

Finally, output verification showed no missing or duplicated rows. All species columns were present and formatted correctly. Inference runtime remained efficient—processing all files took under 20 minutes with batch-wise memory optimization. Garbage collection and CUDA cache flushing were used to prevent memory buildup.

# Discussion

Our inference pipeline was tested on over 9,000 soundscape files. It performed reliably under real-world conditions, producing over 100,000 prediction rows with no errors or format issues. Despite this success, several key challenges remain.

One major limitation is class imbalance. Many rare species—especially endangered amphibians or region-specific birds—appear in only a few samples. The model struggles to learn reliable features for these cases. It often fails to detect them or assigns low probabilities. Since evaluation uses macro-averaging, every species contributes equally. Missing a rare species affects the overall score just as much as a common one. Improving recall for rare classes is critical for better competition performance.

Overlapping vocalizations also create difficulties. Unlike clean single-species recordings used for training, the soundscapes contain multiple birds calling at the same time. Some calls overlap in time and frequency. This makes detection harder. The model must learn to disentangle complex acoustic patterns and assign multiple correct labels per segment. Our current models use sigmoid outputs, but more advanced multi-label architectures may be needed.

Environmental noise adds another layer of complexity. Tropical soundscapes are filled with wind, rain, insects, and background animals. The signal-to-noise ratio is often low. Some bird calls are faint, high-pitched, or similar to ambient sounds. While denoising and bandpass filters help, these are not always sufficient. We observed that even after preprocessing, most predictions remained cautious. Further work is needed to improve robustness under difficult acoustic conditions.

Finally, we face a domainshift between training and test data. Most training clips are short, single-species, and manually segmented. In contrast, test soundscapes are long, continuous, and multi-species. The model must generalize from clean samples to real forest audio. This mismatch lowers accuracy and highlights the need for domain adaptation strategies.

Despite these limitations, the pipeline showed strong internal consistency. Preprocessing, test-time augmentation, ensemble inference, and temporal smoothing all improved stability. No missing data or memory issues occurred. These results confirm the system is ready for deployment, but future work should address rare class recall, multi-label separation, and domain shift. Incorporating attention mechanisms, adaptive thresholds, or pretraining on larger acoustic datasets may help bridge these gaps.

# Reference

* *Adams, S. (n.d.). YouTube. https://www.youtube.com/watch?v=RMfeYitdO-c&t=28s*
* *NIMBioS. (n.d.). Machine learning for the classification of animal vocalizations. YouTube. https://www.youtube.com/watch?v=7BpLybPIkT0&t=18s*
* *Awsaf. (n.d.). BirdCLEF24: KerasCV Starter [Train]. Kaggle. Retrieved May 4, 2025, from* [*Kaggle*](https://www.kaggle.com/code/awsaf49/birdclef24-kerascv-starter-train?utm_source=chatgpt.com)
* *Awsaf. (n.d.). BirdCLEF24: KerasCV Starter [Infer]. Kaggle. Retrieved May 4, 2025, from* [*Kaggle*](https://www.kaggle.com/code/awsaf49/birdclef24-kerascv-starter-infer?utm_source=chatgpt.com)
* *Otegui, J., Ariño, A. H., Guralnick, R., & Braak, K. (2020). Global trends in biodiversity data quality: A quantitative evaluation of GBIF-mediated data. Biodiversity Data Journal, 8, e59202.* [*https://doi.org/10.3897/BDJ.8.e59202*](https://doi.org/10.3897/BDJ.8.e59202)
* *Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller models and faster training (arXiv:2104.00298). arXiv.* [*https://arxiv.org/abs/2104.00298*](https://arxiv.org/abs/2104.00298)
* *University of California Santa Cruz. (2022, December 8). Scientists document 3,000 years of changes to Colombian bird diversity. UCSC News.* [*https://news.ucsc.edu/2022/12/colombian-birds/*](https://news.ucsc.edu/2022/12/colombian-birds/)
* *Fan, M., Hou, Y., Zhang, H., Li, Y., Hu, X., & Pan, H. (2024). Research on bird sound classification method based on deep learning and attention mechanism. Procedia Computer Science, 226, 1605–1614. https://doi.org/10.1016/j.procs.2024.04.170*
* *Salamon, J., & Bello, J. P. (2016). Deep convolutional neural networks and data augmentation for environmental sound classification. In Proceedings of the Detection and Classification of Acoustic Scenes and Events (DCASE) (pp. 1–5). CEUR Workshop Proceedings.* [*http://ceur-ws.org/Vol-1609/16090560.pdf*](http://ceur-ws.org/Vol-1609/16090560.pdf)
* *Greenwood, N. (2012, November 3). Gambler's fallacy and the Monte Carlo fallacy. DataGenetics.* [*http://datagenetics.com/blog/november32012/index.html*](http://datagenetics.com/blog/november32012/index.html)
* *Fayek, H. M. (2016, April 21). Speech processing for machine learning.* [*https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html*](https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html)
* *Practical Cryptography. (n.d.). A guide to mel frequency cepstral coefficients (MFCCs).* [*http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/*](http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/)
* *Dangi, N., & Ramachandran, P. (2024). General-purpose audio representation learning via neural codec pretraining (arXiv:2410.03427). arXiv.* [*https://arxiv.org/abs/2410.03427*](https://arxiv.org/abs/2410.03427)